Data-Mining on GBytes of Encrypted Data

Valeria Nikolaenko

In collaboration with
Dan Boneh (Stanford); Udi Weinsberg, Stratis Ioannidis,
Marc Joye, Nina Taft (Technicolor).

Outline

- **Motivation**
- Background on cryptographic tools
- Linear regression
- Our solution
- Experiments and performance

Motivation

Users

Data

Data

Data mining engine

Data model

Privacy concern!

Engine learns **nothing** more than the model!

Data Mining

- Classification
- Regression : linear regression
- Clustering
- Summarization : matrix factorization
- Dependency modeling

Main challenge: make these algorithms privacy preserving and efficient.

Contribution

- Design of a practical system for privacy preserving linear regression
- Implementation
- Experiments on real datasets

Comparison to state of the art:

- Hall *et al.*'11: 2 days vs 3 min
- Graepel *et al.*'12: 10 min vs 2 sec

Outline

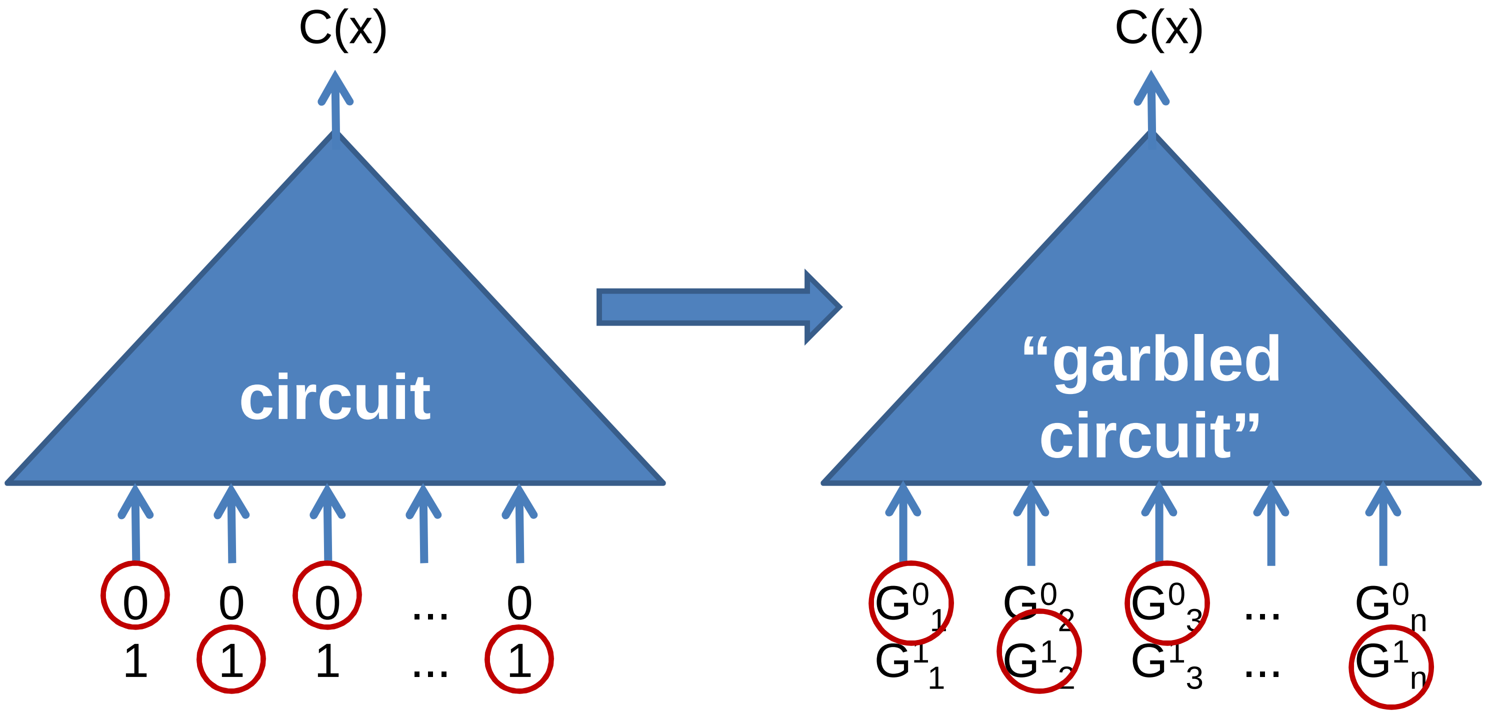
- Motivation
- **Background on cryptographic tools**
- Linear regression
- Our solution
- Experiments and performance

Computations on Encrypted Data

- 2009, **C. Gentry** – FHE
  (slow for our problems)
- 1979, **A. Shamir**
  1988, **BGW**
  Secret sharing
  (huge communication overhead)
- 1982, **A.C. Yao** – Garbled circuits

- Our approach: hybrid of Yao and hom. encryption

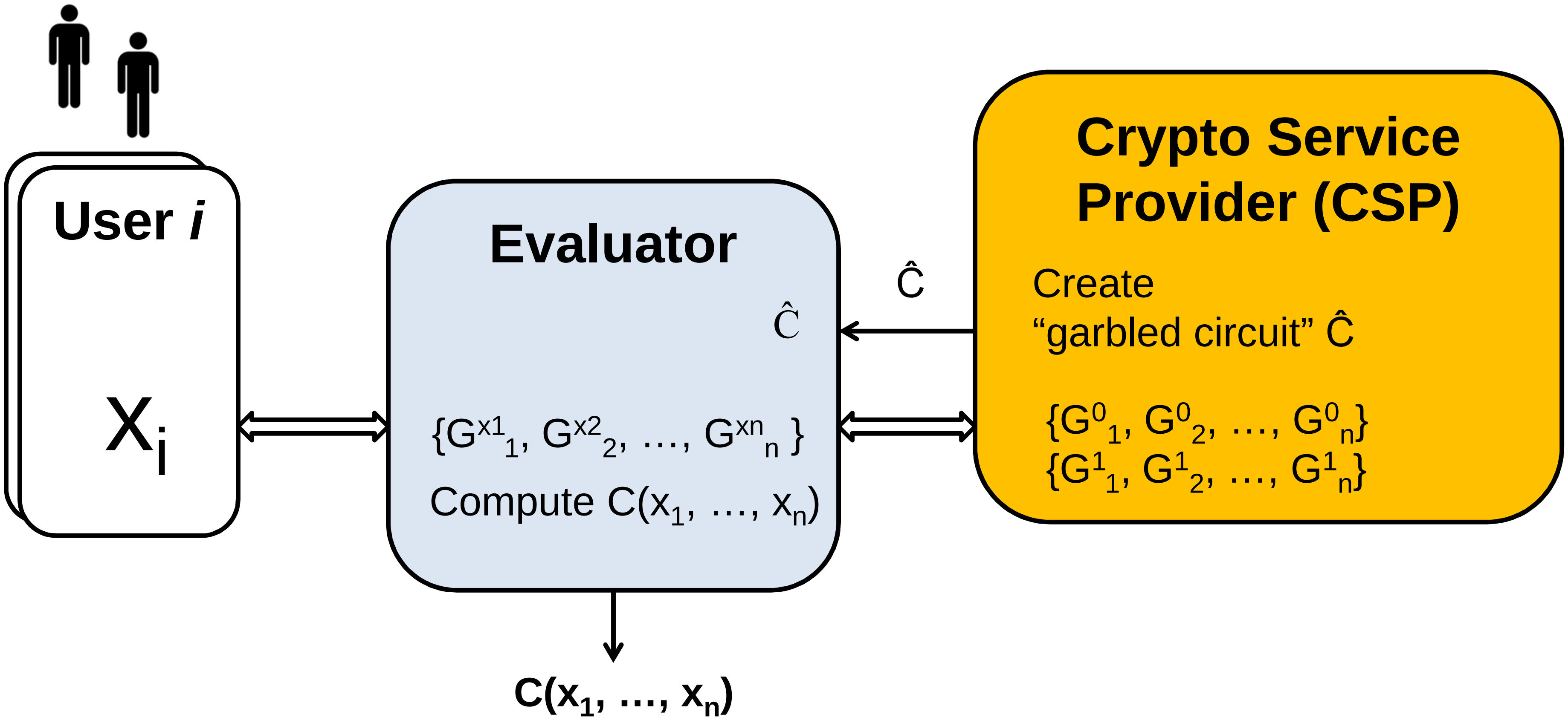
Yao's Garbled Circuits



x=010...1

Nothing is leaked about x, other than C(x)!

Data Mining System Architecture



System Properties

✓ Evaluator learns the model, not the inputs

Problems:

- Not scalable with the number of users
- Users need to be online

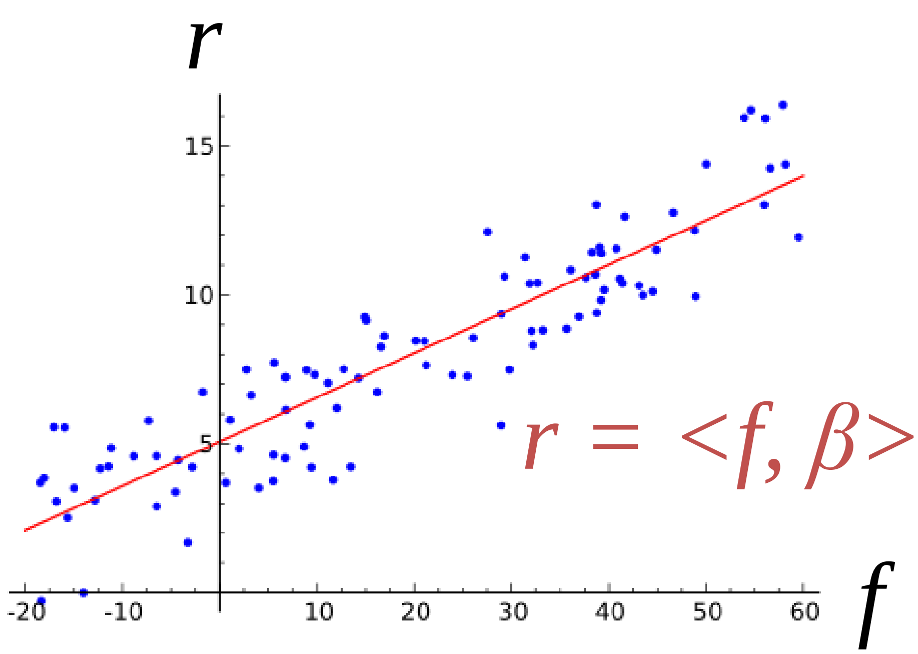
Outline

- Motivation
- Background on cryptographic tools
- **Linear regression**
- Our solution
- Experiments and performance

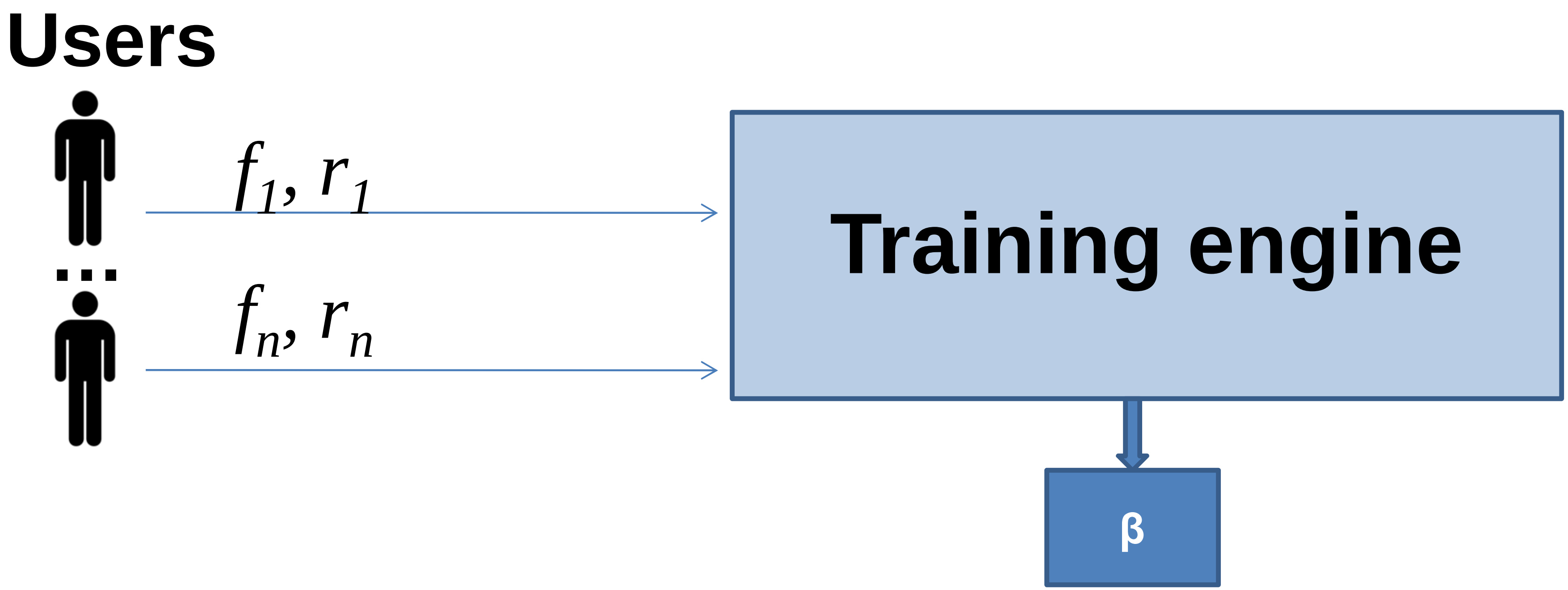
Linear Regression I

- (f,r) – from users
- solve for β

$$A\beta = b$$



Linear Regression II



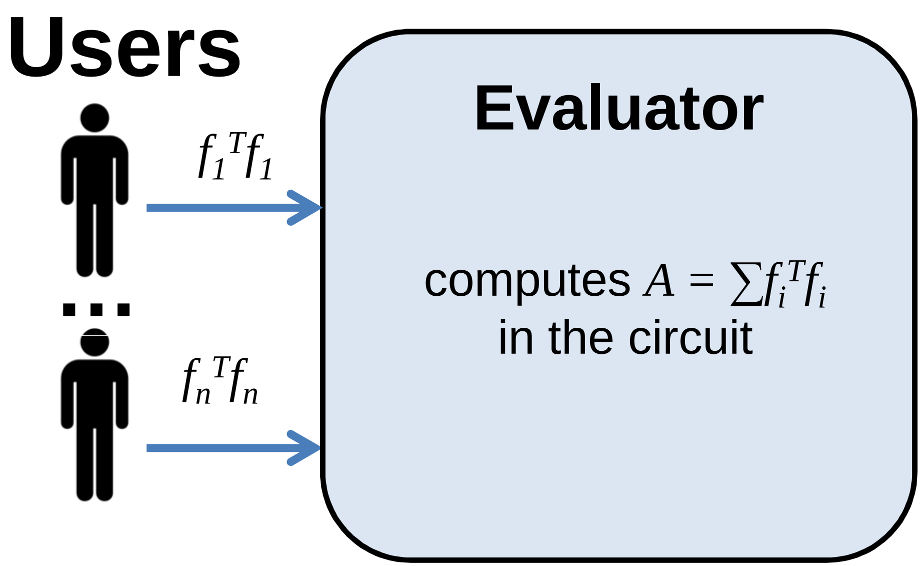
Training engine learns **nothing** about f's and r's, other than β!

Outline

- Motivation
- Background on cryptographic tools
- Linear regression
- **Our solution**
- Experiments and performance

Construct Matrices

Phase1: compute $A = \sum f_i^T f_i$ and $b = \sum f_i^T r_i$

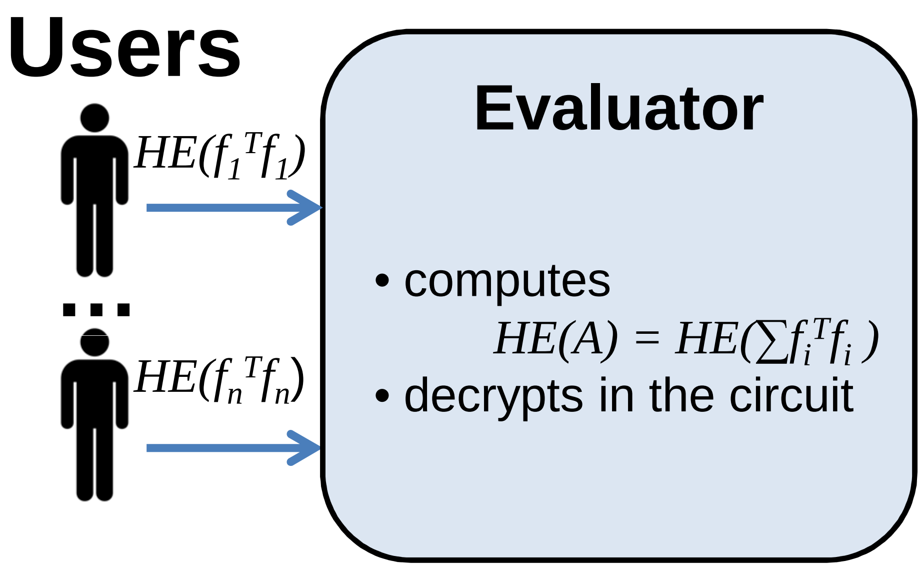


For additions can use <u>homomorphic encryption</u>:

$$[HE(A_1), HE(A_2)] \rightarrow HE(A_1 + A_2)$$

Construct Matrices

Phase1: compute $A = \sum f_i^T f_i$ and $b = \sum f_i^T r_i$



For additions can use <u>homomorphic encryption</u>:

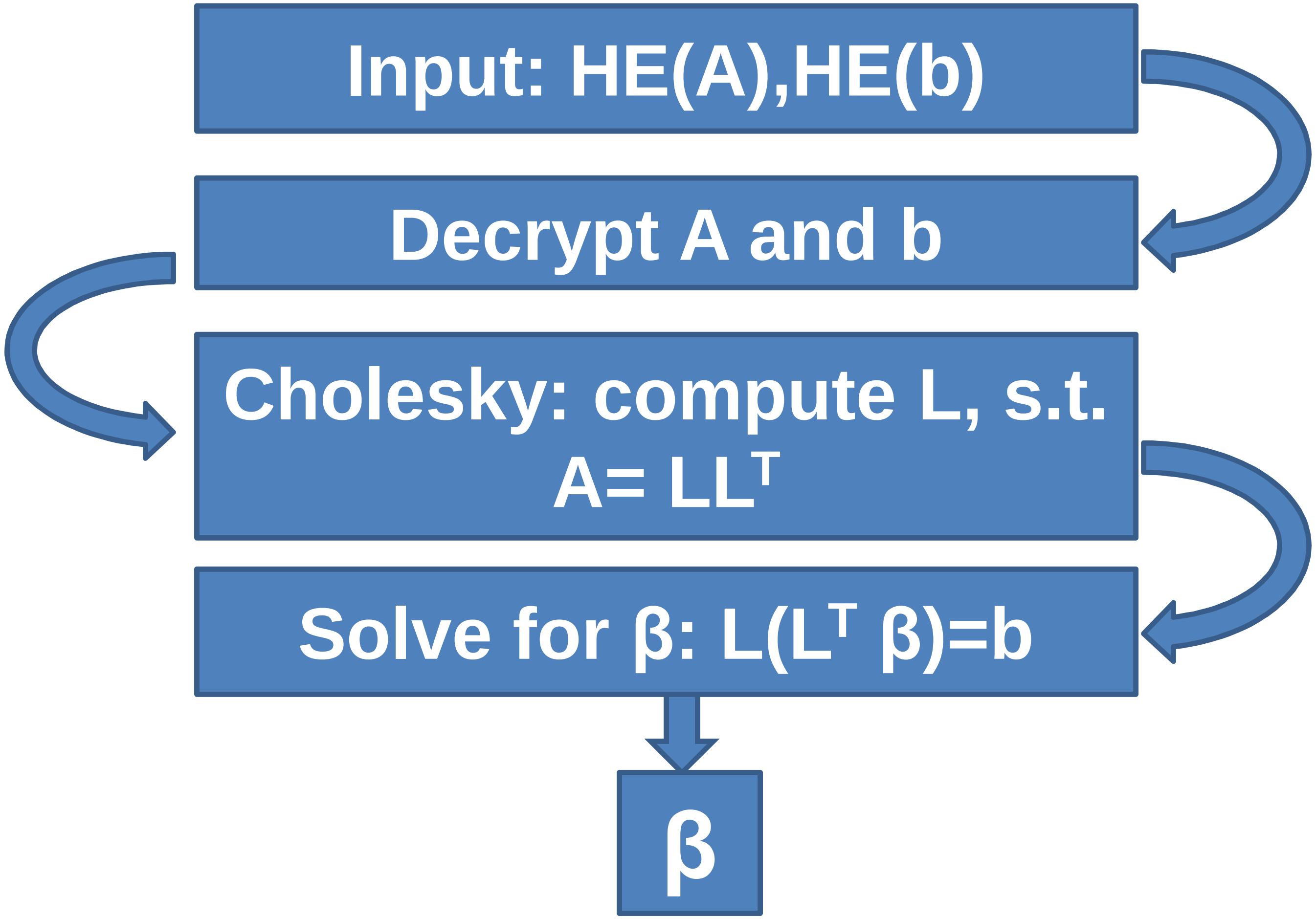
$$[HE(A_1), HE(A_2)] \rightarrow HE(A_1 + A_2)$$

Circuit – independent of the number of users

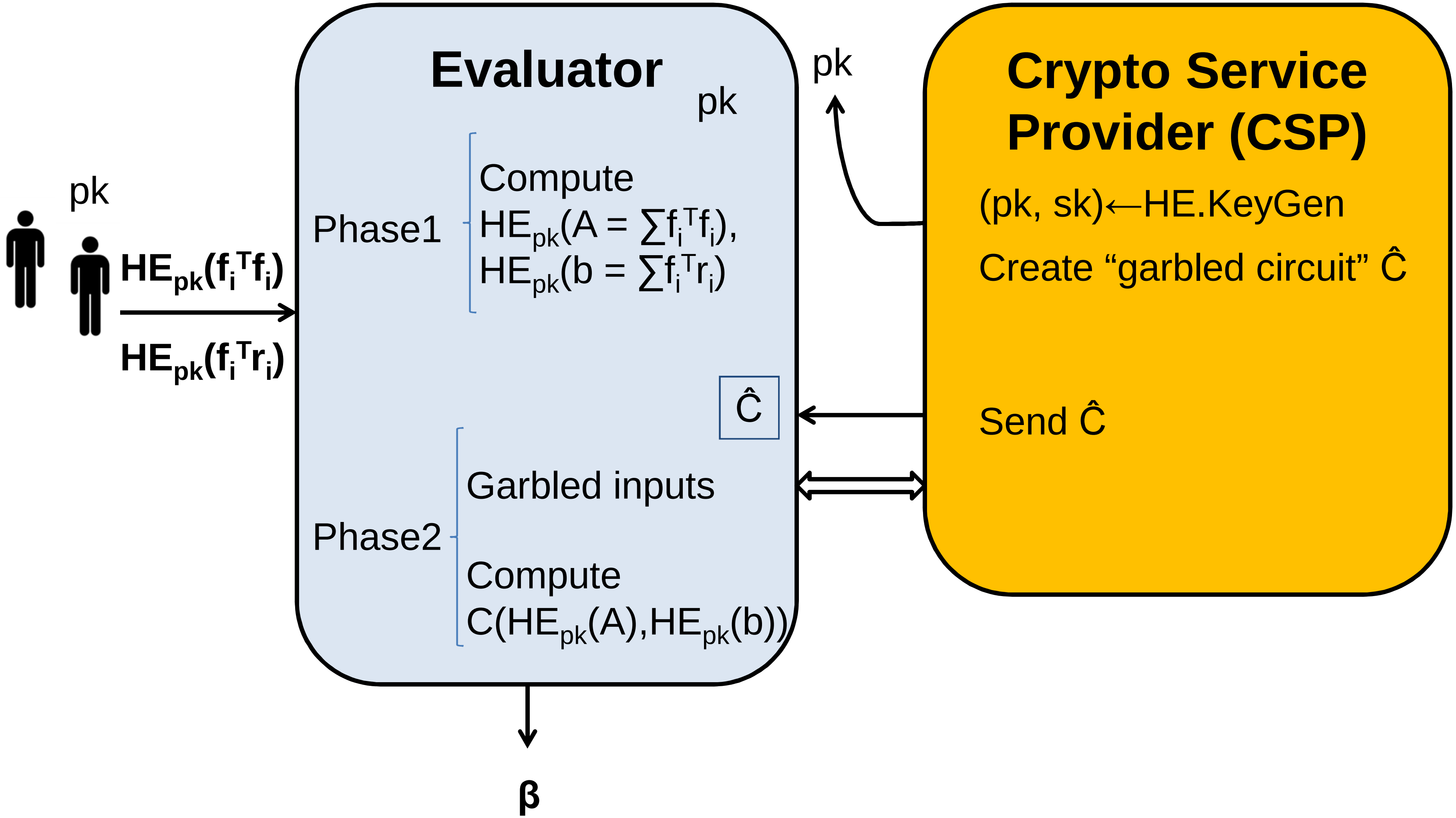
Users – offline

Solve Linear System

Phase2: solve for β $A\beta = b$



Privacy Preserving Regression System



System Properties

✓ Evaluator learns the model, not the inputs
✓ Scalable with the number of users
✓ Users can be offline

Extensions:

- Masking instead of decryption in circuit
- Protection against malicious Evaluator and CSP

Outline

- Motivation
- Background on cryptographic tools
- Linear regression
- Our solution
- **Experiments and performance**

Performance

- Hybrid vs. Pure-Yao: 100 times improvement in time!
- For up to 20 features, 1000’s users
  - time < 3 min
  - communication < 1GB
- For 100 million of users, 20 features: 8.75 hours
- Tested on real datasets

Conclusion

- Privacy-preserving data-mining is **efficient**
- Our approach can be used in practice

- Current work: matrix factorization

- Future work:
  - implement **other data mining algorithms**
  - improving implementation to support **high parallelization**

Thank you!

Questions? valerini@stanford.edu